

EL764085759

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

**Methods and Systems for Retrieving, Organizing, and  
Playing Media Content**

Inventor(s):

Michael Novak

Kipley Olson

Geoff Harris

ATTORNEY'S DOCKET NO. ms1-787us

## **TECHNICAL FIELD**

This invention relates to methods and systems for retrieving, organizing, and playing media content.

## **BACKGROUND**

Today, individuals are able to use their computers to download and play various media content. For example, many companies offer so-called media players that reside on a computer and allow a user to download and experience a variety of media content. For example, users can download media files associated with music and listen to the music via their media player. While users can derive enjoyment from this model, the user experience is really not all that it could be. For example, while users can download music files, they often miss out on a complete user experience such as the one experienced by an individual who purchases a physical CD. This latter individual can now not only listen to their music, but can read song lyrics, learn interesting new facts about the band or artist from the CD jacket, enjoy the CD jacket art, be exposed to information that can facilitate purchasing experiences (e.g. a web site that the user can type into their browser and access), and discover seemingly limitless information relating to their new CD.

In addition, the above described content download model falls short of providing content providers and others, with opportunities to fully market their goods, position their products, and otherwise add value to the end user's experience.

1 Accordingly, this invention arose out of concerns associated with providing  
2 improved methods and systems for organizing, retrieving, and playing media  
3 content.

## 4 5 **SUMMARY**

6 Innovative techniques, systems and methods are described that enable  
7 media content to be packaged and delivered, via a network, in a manner that can  
8 greatly enhance the user experience. A packaging approach provides a  
9 downloadable file that contains different constituent parts that can be processed by  
10 a software-implemented media player to provide a user with not only media  
11 content, but additional content that adds value to the media content. In addition, in  
12 some embodiments, a download approach provides for the downloadable file to be  
13 link-accessed by a user, and automatically downloaded, cataloged, and  
14 experienced by the user without any more user intervention other than clicking on  
15 a particular link that is associated with the downloadable file.

## 16 17 **BRIEF DESCRIPTION OF THE DRAWINGS**

18 Fig. 1 is block diagram of an exemplary system in which various  
19 embodiments can be employed.

20 Fig. 2 is a block diagram of an exemplary server computer.

21 Fig. 3 is a block diagram of an exemplary client computer.

22 Fig. 4 is a diagram of an exemplary media player user interface UI in  
23 accordance with one embodiment.

24 Fig. 5 is a block diagram of an exemplary WMD file in accordance with  
25 one embodiment.

Fig. 6 is a block diagram of an exemplary skin or border file in accordance with one embodiment.

Fig. 7 is a flow diagram that describes steps in a method in accordance with one embodiment.

Fig. 8 is a block diagram of an exemplary computer architecture that can be utilized to implement various embodiments.

Fig. 9 is a diagram that illustrates a portion of an exemplary .asx file in accordance with one embodiment.

Fig. 10 is a flow diagram that describes steps in a method in accordance with one embodiment.

Fig. 11 is a block diagram that is useful in understanding how a WMD file is processed in accordance with one embodiment.

Fig. 12 is a flow diagram that describes steps in a method in accordance with one embodiment.

Fig. 13 is a block diagram that illustrates one aspect of how digital rights management techniques can be utilized in one embodiment.

## **DETAILED DESCRIPTION**

### **Overview**

Innovative techniques, systems and methods are described that enable media content to be packaged and delivered, via a network, in a manner that can greatly enhance the user experience. A packaging approach provides a downloadable file that contains different constituent parts that can be processed by a software-implemented media player to provide a user with not only media content, but additional content that adds value to the media content. In addition, in

1 some embodiments, a download approach provides for the downloadable file to be  
2 link-accessed by a user, and automatically downloaded, cataloged, and  
3 experienced by the user without any more user intervention other than clicking on  
4 a particular link that is associated with the downloadable file.

### 6 **Exemplary System**

7 Fig. 1 shows exemplary systems and a network, generally at 100, in which  
8 the described embodiments can be implemented. The systems can be  
9 implemented in connection with any suitable network. In the embodiment shown,  
10 the system is implemented over the public Internet, using the World Wide Web  
11 (WWW or Web), and its hyperlinking capabilities. The description herein  
12 assumes a general knowledge of technologies relating to the Internet, and  
13 specifically of topics relating to file specification, file retrieval, streaming  
14 multimedia content, and hyperlinking technology.

15 System 100 includes one or more clients 102 and one or more network  
16 servers 104, all of which are connected for data communications over the Internet  
17 106. Each client and server can be implemented as a personal computer or a  
18 similar computer of the type that is typically referred to as "IBM-compatible."

19 An example of a server computer 104 is illustrated in block form in Fig. 2  
20 and includes conventional components such as a data processor 200; volatile and  
21 non-volatile primary electronic memory 202; secondary memory 204 such as hard  
22 disks and floppy disks or other removable media; network interface components  
23 206; display devices interfaces and drivers 208; and other components that are  
24 well known. The computer runs an operating system 210 such as the Windows  
25 NT operating system. The server can also be configured with a digital rights

1 management module 212 that is programmed to provide and enforce digital rights  
2 with respect to multimedia and other content that it sends to clients 102. Such  
3 digital rights can include, without limitation, functionalities including encryption,  
4 key exchange, license delivery and the like.

5 Network servers 104 and their operating systems can be configured in  
6 accordance with known technology, so that they are capable of streaming data  
7 connections with clients. The servers include storage components (such as  
8 secondary memory 204), on which various data files are stored and formatted  
9 appropriately for efficient transmission using known protocols. Compression  
10 techniques can be desirably used to make the most efficient use of limited Internet  
11 bandwidth.

12 Fig. 3 shows an example of a client computer 102. Various types of clients  
13 can be utilized, such as personal computers, palmtop computers, notebook  
14 computers, personal organizers, etc. Client computer 104 includes conventional  
15 components similar to those of network server 104, including a data processor  
16 300; volatile and non-volatile primary electronic memory 301; secondary memory  
17 302 such as hard disks and floppy disks or other removable media; network  
18 interface components 303; display devices interfaces and drivers 304; audio  
19 recording and rendering components 305; and other components as are common in  
20 personal computers.

21 In the case of both network server 104 and client computer 102, the data  
22 processors are programmed by means of instructions stored at different times in  
23 the various computer-readable storage media of the computers. Programs are  
24 typically distributed, for example, on floppy disks or CD-ROMs. From there, they  
25 are installed or loaded into the secondary memory of a computer. At execution,

1 they are loaded at least partially into the computer's primary electronic memory.  
2 The embodiments described herein can include these various types of computer-  
3 readable storage media when such media contain instructions or programs for  
4 implementing the described steps in conjunction with a microprocessor or other  
5 data processor. The embodiments can also include the computer itself when  
6 programmed according to the methods and techniques described below.

7 For purposes of illustration, programs and program components are shown  
8 in Figs. 2 and 3 as discrete blocks within a computer, although it is recognized that  
9 such programs and components reside at various times in different storage  
10 components of the computer.

11 Client 102 is desirably configured with a consumer-oriented operating  
12 system 306, such as one of Microsoft Corporation's Windows operating systems.  
13 In addition, client 102 can run an Internet browser 307, such as Microsoft's  
14 Internet Explorer.

15 Client 102 can also include a multimedia data player or rendering  
16 component 308. An exemplary multimedia player is Microsoft's Media Player 7.  
17 This software component can be capable of establishing data connections with  
18 Internet servers or other servers, and of rendering the multimedia data as audio or  
19 video.

20 Player 308 can be implemented in any suitable hardware, software,  
21 firmware, or combination thereof. In the illustrated and described embodiment, it  
22 can be implemented as a standalone software component, as an ActiveX control  
23 (ActiveX controls are standard features of programs designed for Windows  
24 operating systems), or any other suitable software component.  
25

1 In the illustrated and described embodiment, media player 308 is registered  
2 with the operating system so that it is invoked to open certain types of files,  
3 discussed in more detail below, in response to user requests. In the Windows  
4 operating system, such a user request can be made by clicking on an icon or a link  
5 that is associated with the file types. For example, when browsing to a Web site  
6 that contains links to certain music for purchasing, a user can simply click on a  
7 link. When this happens, the media player can be loaded and executed, and the  
8 file types can be provided to the media player for processing that is described  
9 below in more detail.

### 11 **Exemplary Media Player UI**

12 Fig. 4 shows one exemplary media player user interface (UI) 400 that  
13 comprises part of a media player executing on a client computer. The media  
14 player UI includes a menu 402 that can be used to manage the media player and  
15 various media content that can be played on and by the media player. Drop down  
16 menus are provided for file management, view management, play management,  
17 tools management and help management. In addition, a set of controls 404 are  
18 provided that enable a user to pause, stop, rewind, fast forward and adjust the  
19 volume of media that is currently playing on the media player.

20 A “Now Playing” area or pane 406 is provided in the UI and serves to  
21 enable media-specific content to be displayed for the user. The “Now Playing”  
22 area is highlighted with dashed lines. In the illustrated example, the U2 song  
23 “Beautiful Day” is playing and is accompanied by some visually pleasing art as  
24 well as information concerning the track.



1 In the embodiments about to be described, methods and systems are  
2 provided that can establish a relationship between media content (such as the U2  
3 song “Beautiful Day”), and various other visual and audio content so that when a  
4 user downloads and plays media content, they are automatically presented with the  
5 various other content thereby enriching their experience. Media delivery  
6 techniques and systems are also described that permit and facilitate media delivery  
7 in a manner that can be transparent to the user. In some instances, all the user has  
8 to know how to do is click a link associated with the media content—everything  
9 else is done for them automatically!

### 10 11 **WMD File Overview**

12 In one embodiment, an enhanced user experience is provided through the  
13 use of an innovative file referred to as a “WMD” file for Microsoft “Windows  
14 Media Download.” A WMD file can be thought of as a mechanism that enables a  
15 user to experience a “virtual album” through the use of a single file. Specifically,  
16 a WMD file enables delivery of a whole album’s worth of content – not only just  
17 the music or music files. In addition to the innovative WMD file, a download  
18 process is specifically tailored to enhance the user experience by making the file  
19 acquisition process as easy as possible. A desirable way to do this is to make the  
20 download process a “one-click” process. Hence, a user need only, through the use  
21 of their browser, click a link associated with the WMD file and have the entire  
22 contents of the WMD file downloaded to their computer and organized so that  
23 their media player can automatically begin playing the content and providing the  
24 enhanced user experience.

Fig. 5 shows an exemplary WMD file 500 and its exemplary constituent parts in accordance with one embodiment. In this example, WMD file 500 contains one or more media-specific files 502 that can be used to render a media-specific experience, such as a visual experience, on the media player. Media-specific files (having a .wmz extension) can come in the form of borders or skins. Borders and skins are discussed in more detail below in a section entitled "Borders and Skins". In the described embodiment, borders and skins can be used by individuals, such as content owners, to define a unique user experience when playing media content on a media player. For example, in the Fig. 4 illustration, a border was used to define the content in the "Now Playing" portion of the media player UI in association with playing the U2 song "Beautiful Day."

The WMD file can also contain one or more metafiles 504. Metafiles have an ".asx" extension and are embodied as extensible markup language (XML) files that contain elements or tags that can be used to include information associated with playlists, tracks, borders and skins. XML files and, specifically metafiles, are discussed below in more detail in a section entitled "Metafiles."

The WMD file can also include one or more files associated with media content 506 itself. The media content can include any media content that is supported or supportable by any suitable media player. In the present example, the media content file extensions that are supported by the described media player include, without limitation, .wma, .wmv, .asf, .wav, .avi, .mpg, and .mp3. In addition to the above constituent parts, WMD file 500 can contain other content 508 that can be used in connection with renderable media content. Such other content can include, without limitation HTML, ... The WMD file is embodied, in this example, as a Zip file.

## Borders and Skins

Borders and skins enable the creation of a custom graphical interface for packaged content (i.e. WMD files). Borders and skins can include elements such as images, interactive controls, and links to Web sites. Borders and skins can be used in instances where it is desirable to add additional value to packaged content, such as branding or advertising. In one embodiment, a media player can download the WMD file and then automatically display a skin or border when the packaged content is played.

Borders and skins are different, however, in the interface that is presented. Skins enable individuals to completely customize the interface of a media player, while a border is displayed in certain portions of a media player. In the example of Fig. 4, a border would be displayed in the "Now Playing" area 406. Skins and borders are, however, created using the same tools. These tools are described in U.S. Patent Application Serial Nos. 09/773,446 and 09/773,457, the disclosures of which are incorporated by reference.

Exploring the concept of borders and skins and how they are created in more detail, consider the following. The skinning model described in the above patent applications and in some detail here, permits the creation of skins and borders that are adaptable, dynamic and not constrained in their layout. The skinning model provides all the tools one needs to make a custom user interface. In the context of Windows Media Player, the user interface can include buttons, slider bars, video windows, visualization windows, equalization bars, and so on.

A skin or border is typically composed of several files. Technically speaking, a skin or border comprises a group of files, with each file containing a

1 specific kind of information that is utilized to render or in some way provide  
2 functionality for the skin or border.

3 Fig. 6 shows a diagram that illustrates exemplary file types that can be  
4 utilized to make up or define a skin or border 600. In this example, the exemplary  
5 file types include a skin or border definition file 602, one or more art files 604, and  
6 one or more script files 606. It is to be appreciated and understood that script files  
7 606 may or may not be used to make up a skin or border. The script files, as will  
8 become evident below, provide a means by which a skin can be interactive relative  
9 to various events. Script files can be used to do such things as view the lyrics to a  
10 song as the song plays, or alternate billboards to advertise a product.

11 Skin or border definition file 602 is a master file that defines how the other  
12 files will be used. In the illustrated embodiment, this is a text file and has the  
13 extension “.wms”. Skin definition files are analogous to traffic coordinators of a  
14 skin. Inside this file are the basic instructions for what the skin does and where  
15 the other pieces are. There is typically only one skin or border definition file for a  
16 skin or border. The skin or border definition file and related files can be collected  
17 and compressed into a compressed file, e.g. a Zip file. When this is the case, the  
18 extension is “.wmz” (Windows Media Zipped).

19 Instructions in the skin or border definition file are written in a hierarchical  
20 tag-based language. In the illustrated example, this language is XML, (i.e.  
21 eXtensible Markup Language), an extension to HTML. The XML in the skin or  
22 border definition file uses a set of special element tags or tag pairs to define parts  
23 of the skin user interface. For example, a <BUTTON> tag can define how a  
24 button will behave, where it will go, and what it will look like.

1 Each element tag has specific attributes or properties. In this document, the  
2 terms “attributes” and “properties” will be used interchangeably. For example, the  
3 BUTTON element has an Image attribute that defines where the picture of the  
4 button can be found. This is similar to HTML, where the BODY element will  
5 have a BgColor attribute which defines the background color of the body of the  
6 HTML page.

7 In the illustrated and described embodiment, the skin or border definition  
8 file follows a specific structure. You start with a Theme, create one or more  
9 Views, and then define each View with the user interface elements appropriate for  
10 the type of View you want to use.

11 The Theme element is the root element for a skin. There is only one Theme  
12 element in a skin or border definition file, and it is at the top level. Each Theme  
13 has at least one View. The View governs the particular image you see on the  
14 screen. There may be more than one View to enable switching back and forth.  
15 For example, you might want to have a large view for working with playlists, a  
16 medium view for watching visualizations, and a tiny view that fits in a corner of  
17 the screen.

18 Each View element can also have one or more Subview elements. A  
19 Subview element is similar to a view element and can be used for parts of a skin  
20 that you want to move around, hide, or show. For example, a Subview element  
21 might be used to create a sliding tray that pops out of your skin to display a  
22 graphic equalizer. Subviews are discussed in more detail in the Patent  
23 Applications incorporated by reference above.

24 Once Theme and View elements are defined, the View is populated with  
25 specific user interface elements. Any suitable user interface elements can be used.

1 If an element can be seen by the user, it is called a control. Examples of controls  
2 can include, without limitation, the following controls: Buttons, Sliders, Custom  
3 Sliders, and Progress Bars, Text control, Video Windows, Visualization Windows,  
4 Playlist Windows, and SubView Windows.

5 Each skin or border also has one or more associated art files 604 . In the  
6 illustrated and described embodiment, there are three uses of art or art files in the  
7 skins and borders.

8 First, there are primary images. Primary images are what the users will see  
9 when a skin or border. The primary image is composed of one or more images  
10 that are created by specific skin controls. If you have more than one control, you  
11 can typically specify a "z-order". The z-order defines which controls are  
12 displayed "in front" of other ones. There are also secondary images, such as a  
13 sliding tray, that do not display when a skin first appears, but that show up when  
14 the user takes some action. These follow the same rules as primary images, in that  
15 they are created with a set of controls. Second, there are mapping images.  
16 Mapping images are used for specific controls to specify which regions will  
17 respond to mouse clicks and to determine which controls receive which events.  
18 Third, there are alternate images that can be displayed when a user does  
19 something. For example, you can create an alternate image of a button that will be  
20 displayed only when the mouse hovers over the button. This is a good way to let  
21 users know what they can do, and also allows for a highly discoverable user  
22 interface. The art files can have any suitable type of format. The following  
23 format types are recognized by the Windows Media Player: BMP, JPG, GIF, and  
24 PNG. These uses for art files are discussed in more detail, and specific examples  
25

1 are given in the patent applications incorporated by reference above. For the sake  
2 of brevity, a more detailed explanation and the examples have been omitted.

3 Another type of file or files that can be, but need not be included in the skin  
4 or border definition is a script file(s) 606. In the illustrated and described  
5 example, script files in the form of Jscript files are utilized. It is to be appreciated  
6 and understood that any suitable script types can be utilized. Script files are  
7 typically text files and can be used to create elaborate functionality behind a skin.  
8 By creating functions in JScript, you can do almost anything imaginable with  
9 skins, as will be appreciated by those of skill in the art. For example, you could  
10 use a different playlist for every day of the week, but always have the same one on  
11 Friday.

12 The use of script files provides the capability for a skin or border to respond  
13 to various events. For example, through the use of script files, a skin or border can  
14 "do" something (i.e. react) when the user clicks on a button. Script files also  
15 enable a skin or border to respond to changes that happen to the application which,  
16 in the present case, is Windows Media Player. Such response might be one that  
17 changes, for example, with the progress of the media file that is playing.

18 Fig. 7 is a flow diagram that describes steps in a skin- or border-organizing  
19 method in accordance with one embodiment. The method can be implemented in  
20 any suitable hardware, software, firmware or combination thereof. In the  
21 described embodiment, aspects of the method are implemented in software.

22 Step 700 provides one or more file types that define different aspects of a  
23 skin or border. In the illustrated and described embodiment, the file types can  
24 include, without limitation, file types associated with art (termed "art files" above)  
25 that is used to define aspects of a skin's or border's appearance, and file types

1 associated with script that provide for skin or border interactivity. It will be  
2 appreciated that the art files that can be utilized as described above can extend the  
3 look of a particular skin much further than was previously possible using static  
4 bitmaps. The script files provide the ability to impart functionality and  
5 interactivity to skins that was heretofore entirely absent from conventional  
6 skinning models. Step 702 organizes the file types using a hierarchical tag-based  
7 structure. In the illustrated and described embodiment, this step is implemented  
8 using an XML data structure. An example of a suitable XML data structure is  
9 described in the patent applications incorporated by reference above. The use of  
10 XML to represent or describe a skin or border constitutes a noteworthy and  
11 inventive departure from past methods. In the past, static bitmaps were used to  
12 impart a different *look* to a skin. The layout, feel and functionality of a skin,  
13 however, were fixed. By using XML to describe or define the skin or border, the  
14 skin or border designer is given the flexibility to robustly vary not only the look,  
15 but the layout, feel and functionality as well. Step 704 processes the hierarchical  
16 tag-based structure to provide a skin or border. An exemplary computer  
17 architecture that is capable of processing the hierarchical tag-based structure is  
18 described directly below.

19 The description below constitutes but one exemplary computer architecture  
20 or object model that is capable of implementing the inventive skinning and  
21 bordering techniques and methods. It is to be appreciated that departures from the  
22 described architecture can be made without departing from the spirit and scope of  
23 the claimed subject matter.

24 Fig. 8 shows a computer architecture generally at 800 that comprises a ZIP  
25 reader 802, a XML parser 804, a layout manager 806, a script engine 808, and one



1 or more rendering elements 810. In this example, layout manager 806 is  
2 implemented as an ActiveX control and comprises an object model builder 812  
3 and a rendering engine 814. This architecture is used to process the skin or border  
4 definition file and provide the software elements or components that cooperate to  
5 render a skin or border.

6 Recall that the skin definition file is defined as an XML file that can be  
7 compressed as a WMZ file. The layout manager 806 is given a file name to a skin  
8 or border definition file. The layout manager then instructs zip reader 802 to  
9 extract the XML and any files, including script files, that are included with the  
10 WMZ file. This extraction can be done directly to memory so that the files are not  
11 exploded into separate folders.

12 The skin or border definition file (in XML format) is received and  
13 processed by XML parser 804. Any suitable XML parser can be used. The parser  
14 parses the skin or border definition file to provide an intermediate representation  
15 in the form of a hierarchical data structure that describes the skin and its attributes.  
16 An exemplary hierarchical data structure in the form of a tree is diagrammatically  
17 represented adjacent the XML parser in Fig. 8.

18 The layout manager uses the intermediate representation to create a  
19 scriptable object model. Specifically, in this example, object model builder 812  
20 processes the intermediate representation to provide one or more rendering  
21 elements 810 which are objects in the object model. The rendering elements are  
22 created in memory. To do this, the object model builder simply walks the  
23 intermediate representation looking for the appropriate XML tags, and then creates  
24 the necessary objects or rendering elements that are associated with the tags. The  
25 rendering elements represent components that have been specifically written or

1 defined for the skin. For example, the rendering elements can comprise controls  
2 such as buttons and/or other elements of a skin. Examples of other rendering  
3 elements can include those associated with text, bitmaps, text boxes, lists of text,  
4 pop up menus, and play lists. All of the rendering elements are created by  
5 examining the specification in the XML.

6 If necessary—that is, if the XML specifies script in connection with the  
7 skin or border definition, the object model builder 812 instantiates script engine  
8 808. After the script engine is instantiated, the layout manager provides the  
9 engine with a handle to the various rendering elements that have been created in  
10 memory. This is done so that the script engine can call the various rendering  
11 elements if and when various events require interaction with the rendering  
12 element.

13 Consider the following example: a user defines a skin in the form of a  
14 yellow duck. This skin has three rendering elements—one associated with a head,  
15 one associated with a body, and one associated with the duck's legs. Assume also  
16 that the user wishes for the duck's legs to walk whenever music is played by the  
17 media player. To impart this functionality or interactivity to the skin, the user  
18 authors script that essentially waits for an event associated with music playing.  
19 This event then triggers the script engine to cause the duck's legs to be redrawn by  
20 rendering engine 814 anytime music is playing and in manner that appears as if the  
21 duck is walking. The process for displaying images or art in the "Now Playing"  
22 area 406 of media player UI 400 in Fig. 4 is analogous.

23 Continuing, once the rendering elements (and, if necessary, the script  
24 engine) have been created and any necessary relationships established between the  
25 rendering elements and the script engine, properties for the individual rendering

1 elements are populated. Additionally, any events that are specified in the XML for  
2 any of the rendering elements are provided to script engine 808. To populate the  
3 rendering element properties, in this example, the layout manager 806 processes  
4 the XML intermediate representation to identify various "name-value" pairs. The  
5 name-value pairs can include such things as "visible = true" etc. The layout  
6 manager or, more accurately object model builder 812 filters through all of the  
7 name-value pairs and, for each rendering element 810, it adjusts the properties as  
8 specified in the XML. So, for example, for song lyrics, the object model builder  
9 812 might set a visible property as TRUE when the song is playing.

10 In addition to populating rendering element properties using the XML  
11 intermediate representation, there are also images (i.e. art files and the like) that  
12 are associated with various controls that are referenced in the XML, e.g. the image  
13 that is associated with the down control when it is pressed down. The object  
14 model builder 812 ascertains, from the XML intermediate representation, which  
15 image or art files are needed by the various controls and then passes this property  
16 to the appropriate rendering element. The rendering elements can then call the  
17 layout manager 806 to retrieve the appropriate files that they need.

18 The above-described set up process happens for each of the rendering  
19 elements. That is, the rendering elements are created, their properties are  
20 populated, and they are handed any image or art files or supplemental files that  
21 they need.

22 Once this phase is complete, object model builder 812 can also associate  
23 any script files with their associated skins. The object model accomplishes this  
24 task using the XML intermediate representation. Specifically, there is an attribute  
25 in the XML's VIEW element called scriptFile in which one or more script files

1 can be specified. The object model builder reads this attribute and then requests  
2 the script files from zip reader 802. Accordingly, the script files can be fetched as  
3 needed. Recall that one or more script files were previously placed in memory by  
4 the zip reader and can be obtained from there. Script files can also be fetched  
5 from the ZIP file as needed. Once zip reader 802 provides the requested script  
6 files, layout manager 806 provides them to script engine 808. Accordingly, script  
7 engine 808 now has the code that it needs to impart the scripted functionality to  
8 the button, skin, or border.

9 In one embodiment, skins and borders are inextricably linked with the  
10 media content with which they are associated in the WMD file. Consider again,  
11 Fig 5. There, WMD file 500 comprises media content 506 and media-specific  
12 files 502. The media-specific files 502 comprise the .wmz file that defines the  
13 skin or border described above. The media-specific files and the media content  
14 files (or content) are linked by virtue of a relationship that is defined in the  
15 metafile 504. By virtue of this relationship, any time the media content is played  
16 by the media player, the media-specific files 502 (i.e. the skin or border) can be  
17 automatically rendered and presented to the user. In the case of a skin, this can  
18 entail presenting an entirely new and different media player UI. In the case of the  
19 border, this can entail presenting an entirely new or different *portion* of an already  
20 rendered media player UI. In this manner, when the user plays a particular media  
21 or cause media to be played on their media player, they can be “flipped” into the  
22 associated skin or border. For example, assume that a user is browsing a Web  
23 page and there is a link to a particular piece of content. When the user clicks on  
24 the link, software in the browser notifies the media player, which then  
25 automatically accesses and downloads an associated WMD file. In addition, the

1 media player automatically processes the WMD file and plays any media content  
2 files contained in the WMD file. In addition, if there are any skins or borders  
3 present in the WMD file, the media player causes them to be automatically  
4 displayed for the user. Thus, if the user were looking at their browser UI, the  
5 media player would, in this embodiment, automatically “flip” them to a media  
6 player that contained the skin or the border.

### 8 Metafiles

9 As noted above, a metafile can be used to establish an inextricable link or  
10 relationship between media content and media-specific files such as skin and  
11 border files. This means that when the media content is played by a media player,  
12 the skin or border can be automatically displayed for the user so as to “flip” the  
13 user into a robust viewing/listening experience.

14 Fig. 9 shows an exemplary metafile 900 in the form of an XML file. The  
15 metafile essentially defines a playlist for the media player and references the  
16 media content that is contained in the WMD file. By using XML, the metafile  
17 provides the information that a media player uses to play and display content. The  
18 metafile is made up of various XML elements with their associated tags and  
19 attributes. Each XML element in a metafile defines a particular setting or action  
20 in the media player. A “SKIN” element or tag references a URL to a skin or  
21 border file and is used to display a skin or a border. In this example, the border  
22 file “myborder.wmz” is referenced by URL 902. WMZ files are described above  
23 in connection with Fig. 6. An “ENTRY” element or tag is used to define or  
24 reference specific content that is to be associated with the skin or border specified  
25 by the SKIN tag. In this example, the ENTRY tag contains a URL 904 that

1 references a song file "song.wma." This particular association thus instructs the  
2 media player to download or otherwise retrieve locally, the myborder.wmz skin  
3 file and render it in the "Now Playing" portion of the media player while the  
4 content "song.wma" is playing.

5 Hence, WMD files package a variety of data into a single downloadable  
6 file. For example, users can download a customized WMD file from a Web site.  
7 The WMD file can contain an entire album of music videos that also display  
8 advertising in the form of graphical branding and hyperlinks to an online music  
9 retailer site.

### 10 WMD File Creation

11 In one embodiment, an innovative packaging tool is provided to assist  
12 individuals in packaging data together that can be used to greatly enhance the user  
13 experience when the user plays media on an enabled media player. Exemplary  
14 individuals who can use the innovative packaging tool include, without limitation,  
15 content owners who can now uniquely position their media content in the relevant  
16 consuming market. The packaging tool can be implemented in any suitable  
17 hardware, software, firmware, or combination thereof. In the illustrated example,  
18 the packaging tool is implemented in software.

19 Fig. 10 is a flow diagram that describes steps in a media content packaging  
20 method in accordance with one embodiment. Step 1000 provides one or more  
21 media-specific files. As mentioned above, these media-specific files can comprise  
22 borders and skins. Exemplary techniques for defining borders and skins are  
23 described in the U.S. Patent Applications incorporated by reference above.  
24  
25

1 Step 1002 provides media content. This step is implemented by providing  
2 any suitable files that can be utilized by the media player of interest. Specific file  
3 type examples are given above. Step 1004 provides any other content that might  
4 be desirable to associate with the media content. Step 1006 defines one or more  
5 metafiles that associate the media-specific files and the media content. Exemplary  
6 metafiles can comprise XML files, a specific example of which is given in Fig. 9.  
7 Step 1008 associates the media-specific files, media content, and metafile(s) in a  
8 single downloadable file referred to above as a WMD file. Once the WMD file is  
9 created, it can be uploaded to a Web site so that all a user has to do to access the  
10 file and all its contents is to click a link to it.

#### 11 Accessing a WMD File

12  
13 Fig. 11 shows a diagrammatic representation of how a WMD file can be  
14 accessed in accordance with one embodiment. After it is created, WMD file 500  
15 is uploaded to a Web site managed by a server. Assume now that a user is using  
16 their Web browser 307 to browse the Internet. They click on a hyperlink that is  
17 associated with WMD file 500, or otherwise provide user-input that indicates they  
18 want to access, receive, or purchase media associated with a WMD file. In this  
19 example, the browser 307 ascertains a handler 1100 for the WMD file. The  
20 handler 1100 is notified because it previously registered for notifications whenever  
21 a link associated with a WMD file is clicked from the browser. The handler is  
22 implemented, in this embodiment, as part of media player 308. It is possible for  
23 the handler to be implemented as a standalone component. Handler 1100  
24 automatically downloads the WMD file to the user's hard drive 302. It then  
25 expands the WMD file 500 into its constituent parts and looks for media content

1 files 506 (Fig. 5) which it then registers in a media library 1102 associated with  
2 the player 308. Once the media content files are registered, the media player 308  
3 looks for any metafiles (.asx files) that include track information and border (or  
4 skin) information. The player then plays the content referenced in the metafiles in  
5 the order specified therein, and in conjunction automatically displays any borders  
6 or skins that might be defined for the tracks or collection of tracks.

7 Fig. 12 is a flow diagram that describes steps in a multimedia access  
8 method in accordance with one embodiment. The method can be implemented in  
9 any suitable hardware, software, firmware, or combination thereof. In the  
10 illustrated example, the method can be implemented in software executing on a  
11 client computer.

12 In step 1200, a user clicks a link associated with media content. The media  
13 content can comprise any suitable media content that can be delivered over a  
14 network such as the Internet. In the embodiment described in this document, the  
15 media content comprises songs that can be delivered in digitized form over the  
16 network. The media content is desirably delivered in a single downloadable  
17 package that contains various items in addition to the songs. Examples of such  
18 other items are discussed above and include such things as art, skins, borders, and  
19 the like. In the described embodiment, the single downloadable package is a  
20 WMD file.

21 The link can be provided as part of a Web site managed by a content  
22 provider that sells albums, CDs, or collections of tracks. When a user clicks the  
23 link, step 1202 ascertains a handler for a WMD file associated with the link. In  
24 this step, the WMD handler has previously registered with the user's browser for  
25 notifications when links that correspond to WMD files are selected by the user. In



1 this particular case, the WMD handler comprises part of the media player. Step  
2 1204 invokes the handler (media player) and step 1206 downloads, via the handler,  
3 the associated WMD file to a client computer. The WMD file can be  
4 automatically downloaded and processed on the client computer in any suitable  
5 way. In one embodiment, downloading and processing of the WMD file takes  
6 place automatically without any input or intervention from the user. In this  
7 manner, the constituent parts of the WMD file discussed above can be organized  
8 on the client and the media player can automatically play any associated media  
9 automatically, without any intervention from the user. The automatic playing of  
10 the media also desirably includes retrieving and rendering any additional content,  
11 such as skins, borders, and the like so that the user experience is greatly enhanced.  
12 The following steps illustrate but one way of processing a WMD file so that  
13 downloading and content playing take place without any user intervention.

14 Step 1208 takes the downloaded WMD file and places it in a predetermined  
15 local directory. For example, the media player can place the WMD file in a  
16 "Virtual Album" directory on the client's local hard drive. Step 1210 expands the  
17 WMD file into a folder. This step is implemented by unzipping the WMD file and  
18 placing the constituent files (i.e. media-specific files (.wmz files), metafiles (.asx  
19 files) and media content files) into an appropriate folder. For example, the WMD  
20 file might be unpacked to place the constituent files in a subdirectory of the  
21 "Virtual Album" directory. In step 1212, the media player looks for any media  
22 content files that were contained in the WMD file and registers them by adding  
23 them to its own media library. Step 1214 locates any metafiles (.asx files) that  
24 were delivered with the WMD file. Recall that the metafiles can contain the  
25 playlist for the media content as well as reference any skins or borders that should

1 be displayed with the content. Fig. 9 shows an exemplary portion of an .asx file.  
2 In step 1216, the media player processes the metafile and begins playing media  
3 referenced in the metafile. Contemporaneously, the media player locates any skin  
4 or border referenced in the meta file (step 1218), retrieves the referenced skin or  
5 border (step 1220), and displays the skin or border while the media is playing.  
6 Steps 1216-1222 can take place in a manner in which their effects appear to be  
7 generally simultaneous to the user.

8 It will be appreciated that skins or borders can be locally retrieved by step  
9 1220, or can be retrieved over the Web. It will also be appreciated that the  
10 implementation of step 1222 (i.e. the display of the skin or border) can be different  
11 depending on what is to be displayed. If a skin is to be displayed, then an entirely  
12 new media player UI can be displayed. This new media player UI can have any  
13 form, shape, appearance and the like. If a border is to be displayed, then only a  
14 portion of a standard media player is modified to present the border.

15 Advantages of at least some of the WMD file embodiments discussed  
16 above include that a single-click downloads, extracts, and catalogs multiple files to  
17 user's computer. Additionally, content plays immediately after being downloaded,  
18 without any intervention from the user. This is advantageous from the standpoint  
19 of making it easier for users to acquire and enjoy media content. Users no longer  
20 have to know anything about the downloading process, e.g. where to place files,  
21 what files to download, how to provide those files to the proper media player and  
22 the like. This can all be done automatically for them. Further, customized borders  
23 and skins provide an opportunity for displaying advertising and branding  
24 information. This greatly enhances the value that can be provided by content  
25 owners. Packaged playlists are automatically cataloged in a media player's library

1 so that a user can make use of them almost immediately. Additionally, a user can  
2 be redirected to a Web site from within a media player to a site that is related to  
3 the content that is being played, i.e. to a Web site for a purchasing experience.  
4 Furthermore, multiple audio and video file types are supported for packaging and  
5 multiple files can be downloaded in one package thus making for a more rich and  
6 robust user experience.

### 7 8 **Digital Rights Management**

9 Recall from the earlier discussion that the content within a WMD file can  
10 be managed through the use of digital rights management techniques. This takes  
11 place through the use of digital rights management modules that can reside on one  
12 or both of a server that provides the WMD files, and a client that receives the  
13 WMD files.

14 Digital rights management (DRM) is a type of server software developed to  
15 enable secure distribution, and perhaps more importantly, to disable illegal  
16 distribution of paid content over the Web. DRM technologies are being  
17 continually developed as a means of protection against the online piracy of  
18 commercially marketed material, which has proliferated through the widespread  
19 use of such things as peer-to-peer file exchange programs. In general, DRM  
20 products are typically turnkey packages that include everything needed for the  
21 operation, such as, for example, server software and user plug-ins.

22 Fig. 13 illustrates multiple different ways in which digital rights  
23 management techniques can be used to manage a WMD file 1300 and content  
24 within the WMD file. The techniques about to be described can be used by  
25 themselves or in various combinations with these and other DRM techniques.

1 To begin with, DRM techniques can be used to protect the WMD file itself.  
2 One exemplary way of doing this is to encrypt the WMD file and associate a  
3 WMD file key 1310 with it. The WMD file key 1310 can be used to decrypt the  
4 WMD file and gain access to the file's contents. Only authorized users can be  
5 given access to the WMD file key and the key itself can be non-transferrable to  
6 other computers. Thus, for example, the WMD file key might itself be encrypted  
7 with a user's public key and sent to the user who properly purchases the WMD  
8 file. Then, by using the user's private key, the media player can decrypt the  
9 encrypted WMD file key so that the WMD file can itself be decrypted. By  
10 encrypting the WMD file, the file can then be used without extracting it into its  
11 constituent parts.

12 In addition to or in place of the above-described technique, the various  
13 separate constituent parts of a WMD file can be protected through DRM  
14 techniques. One way of doing this is to encrypt the content within the individual  
15 constituent parts with a key that is provided only to authorized purchasers.  
16 Provision of the key can take place substantially as described above.

17 Consider, for example media-specific files 1302. These files can represent  
18 great value to the content provider insofar as enhancing the user's experience. In  
19 order for an enhanced experience, (e.g. being able to view lyrics, trivia, art and the  
20 like) associated with particular media content, the user might be required to pay an  
21 additional fee. In this case, when the user pays the additional fee, they are  
22 provided with one or more keys that can be used by their media player to unlock  
23 content provided by the media-specific files. So, for example, a user might be able  
24 to additionally purchase a new border or skin that provides art and interactive  
25 capabilities that can be experienced while they are listening to their new songs.

1 Consider additionally metafiles 1304. Here, DRM techniques can be used  
2 to prevent unauthorized users from gaining access to the .asx files that contain the  
3 data used by the media player to organize and render content. Consider further the  
4 media content 1306 itself. DRM techniques can be used to protect the content  
5 from unauthorized users. Such techniques can also be used to implement pre-  
6 delivered and progressive licenses. Specifically, individual content files within the  
7 media content 1306 can be encrypted with different keys. A user might pay one  
8 fee to access keys that allow them to decrypt a first set of content (e.g. the first 10  
9 tracks of a CD), and then pay an additional fee to access an additional key that  
10 allows them to decrypt two additional bonus tracks.

11 Consider further that any other content 1308 that can be provided within a  
12 WMD file can be protected through the use of DRM techniques.

13 It is to be appreciated and understood that DRM techniques vary widely  
14 and that the above-described exemplary techniques are not intended to limit the  
15 DRM techniques that can be employed with the inventive WMD files.  
16 Accordingly, any suitable DRM techniques can be utilized without departing from  
17 the spirit and scope of the claimed subject matter.

## 18 19 **Conclusion**

20 The embodiments described above enable media content to be efficiently  
21 packaged and delivered, via a network, in a manner that can greatly enhance the  
22 user experience. A packaging approach provides a downloadable file that contains  
23 different constituent parts that can be processed by a software-implemented media  
24 player to provide a user with not only media content, but additional content that  
25 adds value to the media content and enriches the user's experience. A download

1 approach provides for the downloadable file to be link-accessed by a user, and  
2 automatically downloaded, cataloged, and experienced by the user without any  
3 more user intervention other than clicking on a particular link that is associated  
4 with the downloadable file.

5 Although the invention has been described in language specific to structural  
6 features and/or methodological steps, it is to be understood that the invention  
7 defined in the appended claims is not necessarily limited to the specific features or  
8 steps described. Rather, the specific features and steps are disclosed as preferred  
9 forms of implementing the claimed invention.